**Amendments to the Specification:**

Please replace paragraphs [0041], [0042], [0043], [0051], [0060], [0065], [0076], and [00106] with the following amended paragraphs:

**[0041]** The gating circuit 310 essentially performs a multiplication of the input samples with the sine and cosine values without using an actual multiplier. As discussed earlier, the input samples are one-bit samples clocked at the sampling clock signal having a frequency of $16*f_0$. The input samples are then multiplied by two-bit sine and cosine values at $4*f_0$. This multiplication is equivalent to multiplication by two consecutive +1's and two –1's. Furthermore, since the input samples are one-bit having logic values of 0 and 1, the multiplication can be efficiently performed by an exclusive OR operation. The gating circuit includes an exclusive OR gate 312 and a frequency divider 314. The exclusive OR gate 312 performs the multiplication between the input samples and the sine/cosine values of two consecutive +1's and two –1's. The sine and cosine values can be encoded as +1's and 0's. The frequency divider 314 generates the sine/cosine values of +1's and 0's. The frequency divider 314 can be implemented as two flip-flops connected in cascade to effectively divide the sampling clock by four to provide a clock signal having a frequency of $4*f_0$. The frequency divider 314 also generates two complementary clock signals at frequency of $8*f_0$: one is <u>in</u> true form and one is in complementary form. These two complementary $8*f_0$ clock signals are used by the demultiplexer 320. The gating circuit 314 generates a gated input sample at the frequency of the sampling clock signal.

**[0042]** The demultiplexer 320 essentially splits the mixed input samples into in-phase and quadrature components. This is performed by demultiplexing the gated input sample into two signals. The demultiplexer 320 includes two synchronizers 322 and 324. The synchronizers 322 and 324 synchronizes the gated input sample by the true and complementary <u>forms</u> ~~form~~ of the $8*f_0$ clock signal, respectively, to generate the in-phase and quadrature samples. Since the synchronizers 322 and 324 are clocked by the $8*f_0$ clock signal, they essentially down sample the gated input samples at $8*f_0$ frequency by a factor of two.

**[0043]** The integrator/decimator 330 further down samples or decimates the in-phase and quadrature samples by integrating them in an integration interval. The integration interval is selected to be equivalent to a four-sample interval so that the down sampling brings ~~bring~~ the gated input samples to $2*f_0$ samples/sec. This can be done effectively by counting the number of 1's in the gated input sample in a 4-sample interval. The integrator/decimator 330 includes two K-bit counters 332 and 334 to count the number of 1's in the in-phase and quadrature samples from the synchronizers 322 and 324, respectively. The K-bit counters 332 and 334 are reset by a reset signal generated from the reset circuit 350. This reset signal is to start a new integration interval. The K-bit counters 332 and 334 generate in-phase and quadrature decimated samples, respectively, to the mapper 340. K is selected to ensure that the count value can cover the possible range of numbers. Note that the in-phase or quadrature sample is one-bit. Therefore, the possible numbers of bit 1's in the in-phase or quadrature sample in a 4-sample integration interval are 0, 1, 2, 3, and 4. If +1's and –1's are used to encode the integrated samples, these numbers are equivalent to –2, -1, 0, +1, and +2. To represent these numbers, K would have been 3. In other words, the effect of down sampling is that each sample becomes a three-bit sample.

**[0051]** The code register 440 stores M PN code samples transferred from one of the N storage elements $452_1$ to $452_N$. The N storage elements $452_1$ to $452_N$ store N PN code sequences corresponding to N satellites. All N storage elements $452_1$ to $452_N$ operate in synchrony. The N PN code sequences come from the PN generator and re-tracking circuit 240 (Figure 2). The writing of the N PN code samples into the N storage elements $452_1$ to $452_N$ is synchronized with the circular shifting of the circular shift register 420 so that correct phase values are correlated with correct code samples. For each storage element, a code sample is written into the storage element at a code position corresponding to a data position of the corresponding demodulated sample in the circular shift register 420. This writing is essentially equivalent to shifting ~~shift~~ the N storage elements $452_1$ to $452_N$ synchronously with the circular shifting of the circular shift register 420. By avoiding using N shift registers to shift N PN code sequences, significant reduction of power consumption is achieved. The synchronous operation between the circular shift register 420 and the N storage elements $452_1$ to $452_N$ is explained more in Figure 6. The N storage elements $452_1$ to $452_N$ may be implemented as N rows of flip-flops where each row has

M flip-flops, N registers with selectable data write, or N rows of a static random access memory (RAM), or any other suitable storage devices.

**[0060]** At time t+2 Δt, the circular shift register 420 circularly shifts the 22 samples so that each sample is shifted to the left and the leftmost ~~left most~~ sample $a_{n+10}$ is shifted to occupy the rightmost position. At the same time, the writing circuit 460 (Figure 4) writes to the next rightmost position of the N storage elements $452_1$ to $452_N$. Then the code register is loaded with one of the N storage elements $452_1$ to $452_N$. The remaining code samples remain the same. The process continues when all M code samples are written into the N storage elements $452_1$ to $452_N$. At time t+kΔt, N code samples are written into the code position k while the other code positions contain the same code samples. Thereafter, a new sequence of M demodulated samples is transferred to the circular shift register 420 and the process repeats.

**[0065]** The demodulated samples are encoded to have representations of 01, 10, and 11, corresponding to –1, 0, and +1, respectively. The PN code sample is represented by 0 and 1, corresponding to –1 and +1, respectively. The mapper M maps the product to 1, 2, and 3, corresponding to –1, 0, and +1, respectively. The adder produces ~~produce~~ a sum in the range of {+22, +66}. The subtractor subtracts a bias value of 22 from the sum so that the result has a range of {0, ~~+44~~}. This range can be represented by a 6-bit result.

**[0076]** The G2 coder 860 includes a shift register 862 and an exclusive OR gate 864. The shift register 862 has ten elements and shifts the code bits to the right. The shift register 862 is clocked by the PN clock signal k from the corresponding accumulator $820_k$ (Figure 8A). The processor 150 initially loads the shift register 862 with all 1's. The exclusive OR gate 864 is a six-input exclusive OR gate which performs ~~perform~~ an exclusive OR operation on elements 2, 3, 6, 8, 9, and 10 of the shift register 862. The output of the exclusive OR gate 864 goes back to the input of the shift register 862.

**[00106]** Figure 13 is a diagram illustrating the epoch processing circuit 260 shown in Figure 2, which controls the I and Q [[d]] memory circuits according to one embodiment of the invention. The epoch processing circuit 260 includes in-phase and quadrature memory circuits 1310 and 1320, an epoch control circuit 1330, and a multiplexer 1340.